



## First lesson with Scratch

This basic module provides an introduction to programming and the visual programming language Scratch. After a warm-up, which clarifies important programming concepts, the surface of Scratch is explained and the students can explore it in teamwork.



### Time

45 minutes



### Target group

Grade: 4–8



### Level

Beginner



### Prior knowledge

None



### Example video

<https://youtu.be/VKyM8xdU3CM>




### Competences

The students:

- > know the basic concepts of programming.
- > recognize algorithms and can reproduce them.
- > know the surface and the most important functions of Scratch.





### Material

- > Tablet / computer (1 x per team of 2)
- > Internet
- > Projector / Interactive Whiteboard
- > Pen and paper (for the warm-up)
- > Scratch Flashcards (1 x per team of 2) 


## First lesson with Scratch

# Preparation of the lesson

- > One computer or tablet with internet access is required for each team of two.
- > Have pen and paper ready (alternatively, a drawing or note-taking app can be used).
- > Connect a projector or interactive whiteboard and open the website [scratch.mit.edu](https://scratch.mit.edu). 
- > Print out the [Scratch flashcards](#) once per team of 2. 




## Download and video material on the project


 Download material (PDF)  Website  Video

-  [https://coding-for-tomorrow.de/download/CFT\\_Scratch\\_Lernkarten\\_Grundlagen.pdf](https://coding-for-tomorrow.de/download/CFT_Scratch_Lernkarten_Grundlagen.pdf)
-  <https://scratch.mit.edu/projects/editor/>
-  [https://coding-for-tomorrow.de/download/CFT\\_Analoges\\_Programmieren\\_Warmup\\_Algorithmisches\\_Zeichnen.pdf](https://coding-for-tomorrow.de/download/CFT_Analoges_Programmieren_Warmup_Algorithmisches_Zeichnen.pdf)
-  <https://coding-for-tomorrow.de/scratch/>
-  (Bewegung von Objekten) <https://youtu.be/4g-ZHlloIX0>
-  (Aussehen z.B. Sprechblasen hinzufügen) <https://youtu.be/J3EqzLMgO8Y>
-  (Ereignisse z.B. Nachrichten senden und empfangen) <https://youtu.be/howslEbGPiw>
-  (Steuerung z.B. Warte- oder Wiederhole-Befehle hinzufügen) <https://youtu.be/DTZNA5vZAmY>
-  [https://coding-for-tomorrow.de/download/CFT\\_Methode\\_Pair\\_Programming.pdf](https://coding-for-tomorrow.de/download/CFT_Methode_Pair_Programming.pdf)

## Course of the lesson


### Introduction (ca. 10 minutes)

1. Presentation of the goal: Getting to know the Scratch surface and programming a first animation.
2. Execution of the warm-up [algorithmic drawing](#) to provide an initial understanding of algorithms. 
3. Explain the most important functions of the interface or show the students the introductory [videos](#). 
4. Hand out the [Scratch flashcards](#) and give the students the following task: 

 Opens the website [scratch.mit.edu](https://scratch.mit.edu). Get to know the interface and the programming of Scratch by processing the Scratch flashcards. You have 25 minutes for this.



### Hints

-  The students are not allowed to close the tab, otherwise they will lose their project. It is advisable for the students to create an account in order to save their projects online.

**Activity** (ca. 25 minutes)

5. The students get together in teams of two and explore the Scratch interface with help of the flashcards. [↓](#)  
For the collaboration the method [Pair Programming](#) is suitable. [↓](#)

**Wrap-up** (ca. 5 minutes)

6. Make sure all students understand the main features of Scratch:
- a) [Motion of objects](#) [▶](#)
  - b) [Look \(e.g. speech bubbles\)](#) [▶](#)
  - c) [Events \(e.g. broadcast and receive messages\)](#) [▶](#)
  - d) [Control \(e.g. wait or loops\)](#) [▶](#)

**Outlook** (ca. 5 minutes)

7. Show the students more project ideas on the topic of Scratch.